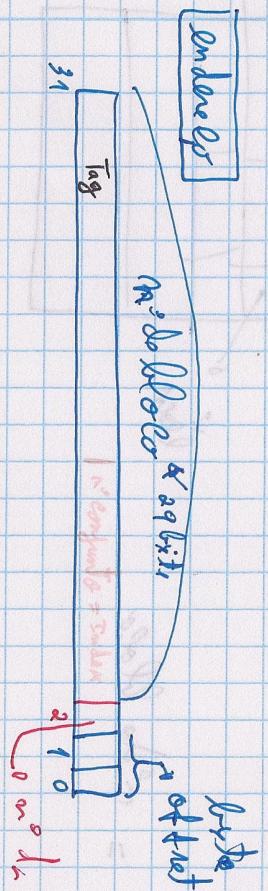


Onderección



$$n^{\circ} \text{ conmutador} - 2^k = k + 1 \text{ bits}$$

nº de bloco % nº conmutador

Resta
divisores

→ igual → ai conmutador
index

index = nº de bloco % nº conmutador

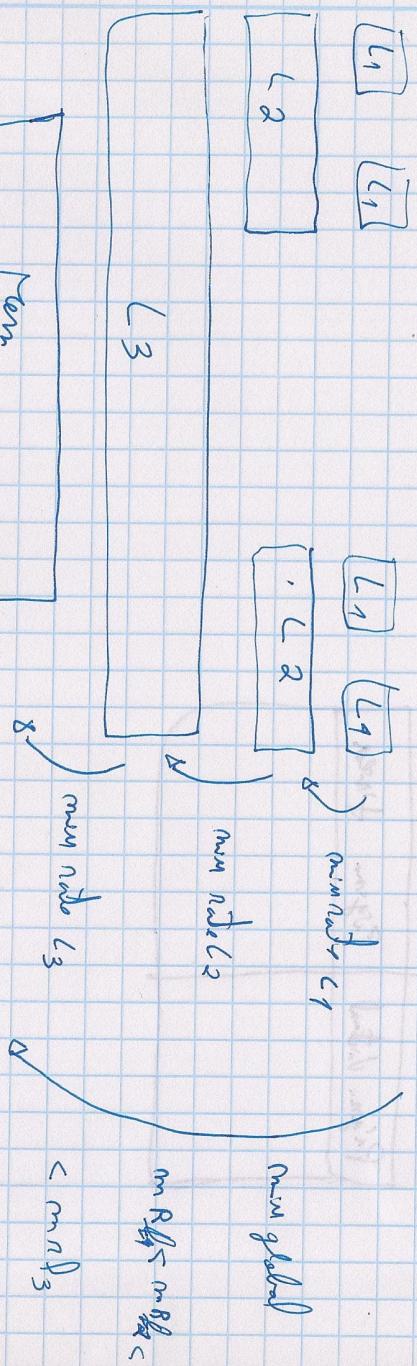
$$\text{tag} = \frac{n^{\circ} \text{ bloco}}{n^{\circ} \text{ conmutador}}$$

(Características de una Cache)

Cache

CPU 1

CPU K



$$\text{min rate } L_1 = \frac{\text{N}^2 \text{ min rate } L_2}{\text{N}^2 \text{ níveis de cache } L_1}$$

Program - 4KB
16KB

$$\text{min rate } L_2 = \frac{\text{N}^2 \text{ min rate } L_3}{\text{N}^2 \text{ níveis de cache } L_2}$$

Page fault - Número que

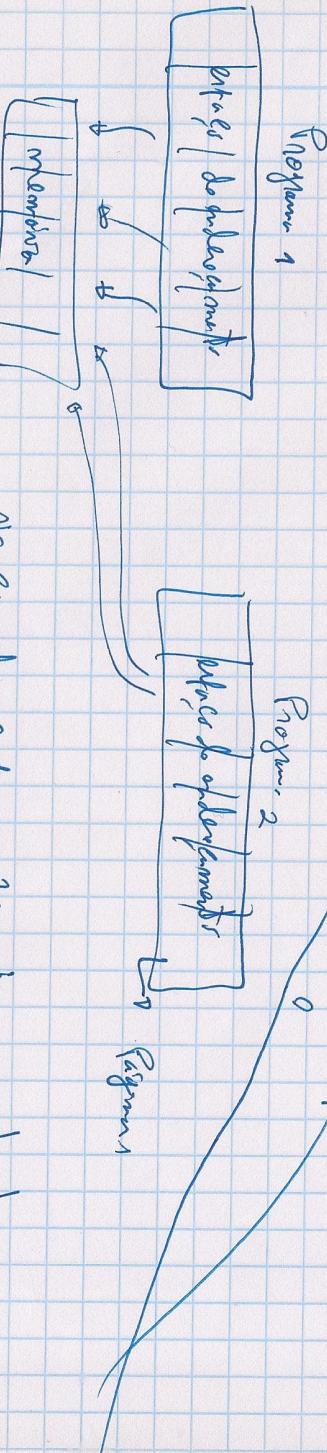
o Cálculo do Programa é menor

mas encontra a Page que contém

uma exceção. O Programa continua

a interromper-se

o processo de busca é reiniciado



No caso de o Programa ~~desconhecer~~ necessitar de

mais endereços, mas não tem mais endereços

no seu domínio ou se não tiverem mais espaço

também informarão de 2 programas

que devem informá-lo que não tem mais endereços

e os dados só serão fornecidos quando os

Notas: **Destino**

→ Tabelas de Páginas = paginas tabela

Pergunta: Calcular o tamanho da num. Tabelas de Páginas imposta se a máquina tiver 2 GB de memória com pag de 8 kB e cluster de 4 kB.

$$\frac{2 \text{ GB}}{8 \text{ kB}} \times 6$$

Nº de entradas

aluno . (sw)

Caché

Memória

Onde é feita a diferença?

- no cache ou na memoria - write through - write back

- write on cache

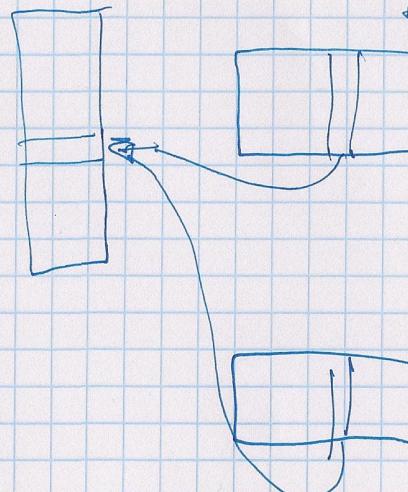
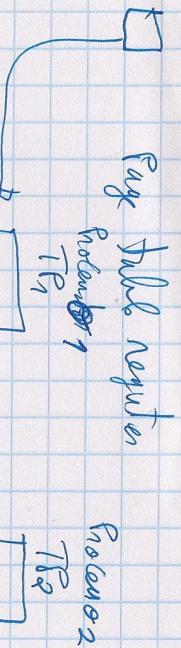
- write back

- escrita na memoria quando bloco é desejado

- dirty bit

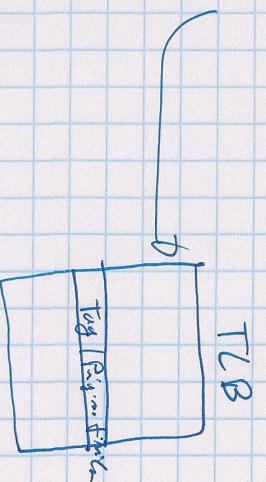
- write back buffer

26 Novembre 2015



Con a cada uno se forman páginas o sección de memoria más segmento

End virtual



TLB miss exception

• Cambia a controlador de tablas de páginas de procesos de pag Virtuales

• Alrededor de TLB

• No conecta

a la memoria física.

① se lleva al TLB

hit
miss \rightarrow TLB miss
excepción

→ tiene un gran diseño magnífico

• Tiempo de acceso = Readd Time + Latencia de controlador + Transf. de datos + Latencia de controlador

$$\text{Latencia de controlador} = \frac{1}{2} \times \frac{60}{\text{Velocidad de reloj}}$$

5.11.1

dem TLB

$$4669 = \underbrace{\dots}_{[0-11]} \underbrace{0001}_{\text{Page offset}} \underbrace{10001101}_{\text{Page address}}$$

$$\text{Page de } 4 \text{ KiB} = \underbrace{2}_{\text{P. offset}} \underbrace{\text{Ht. TLB}}_{\text{Ht. offset}}$$

[5.11.1] Page offset

Indirect (Byte)	4669	2227	13916	34587	48870	12608	49225
-----------------	------	------	-------	-------	-------	-------	-------

Page Virtual	1	0	3	8	11	3	12
--------------	---	---	---	---	----	---	----

Page Table	Page fault	H.t.	Page fault	H.t.	Page fault	H.t.	Page fault
------------	------------	------	------------	------	------------	------	------------

Page Table	13	5	6	14	12	6	15
------------	----	---	---	----	----	---	----

Indirect & red. + offset	13x43 + + offset	5x48 + + offset	6x48 + + offset	14x48 + + offset	12x48 + + offset	6x48 + + offset	10x48 + + offset
-----------------------------	---------------------	--------------------	--------------------	---------------------	---------------------	--------------------	---------------------

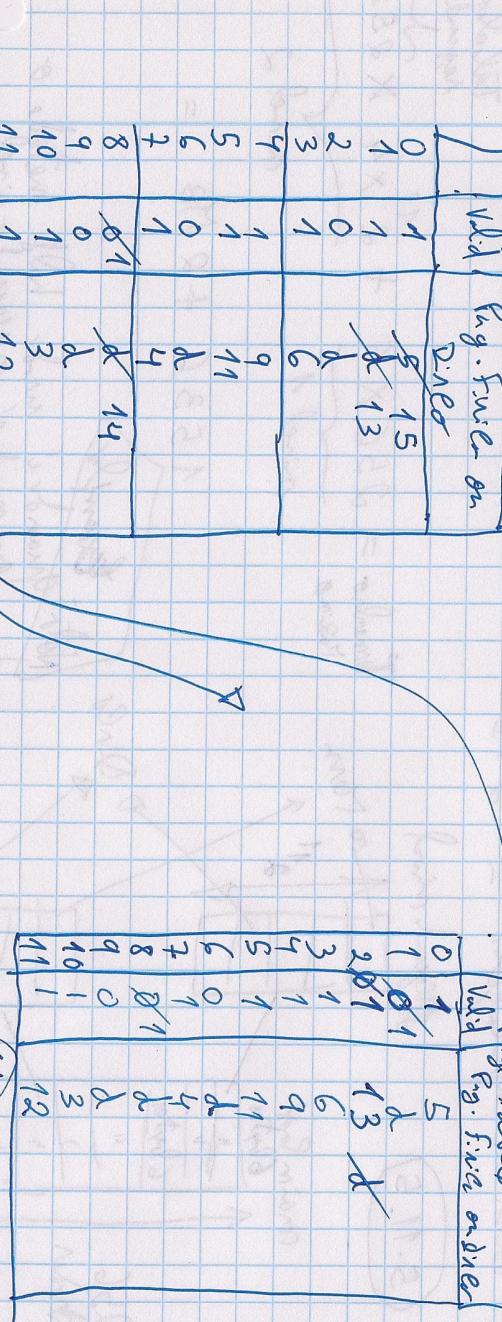
Page Table	0	1	5	1	5	1	5
------------	---	---	---	---	---	---	---

Page Table	1	0	1	3	d	13	d
------------	---	---	---	---	---	----	---

Page Table	0	1	5	13	13	5	6
------------	---	---	---	----	----	---	---

5.11.2

Page de 46 kB = 2

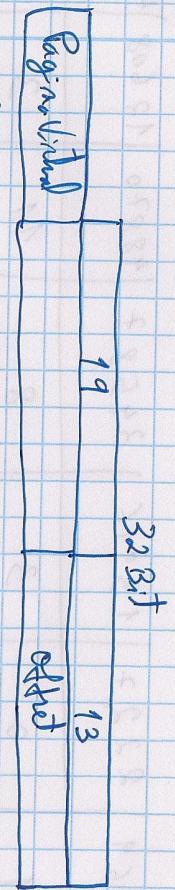


Indirect (Byte)	4669	2227	13916	34587	48870	12608	49225
Page Virtual	6	0	0	2	2	0	3
Page Table	num.TLB	H.t.TLB	H.t.TLB	Page fault	H.t.TLB	H.t.TLB	H.t.TLB
Page Fault	5	5	5	13	13	5	6
Indirect & red. + offset	5x16 kB + offset	0	11	13x16 + offset	0	11	5x16 kB + offset

TLB	V	Tag	Page
1	1	2	12 13
1	2	4	5
1	3	6	5
1	4	0	5

5.11.4

$$\text{Page Nage} = 8 \text{ Kib} = 2^{13} \rightarrow \text{bit offset}$$



$$2^{19} = \text{nr total de page table, fior pagina Virtual} \\ 2^{19} = 524,286 \text{ K}$$

$$524,286 \times 4$$

$$2^{14} = 2 \text{ MB}$$

virtual address

$$\text{Total an apl. pag} = 2 \text{ MB} \times 5 = 10 \text{ MB}$$

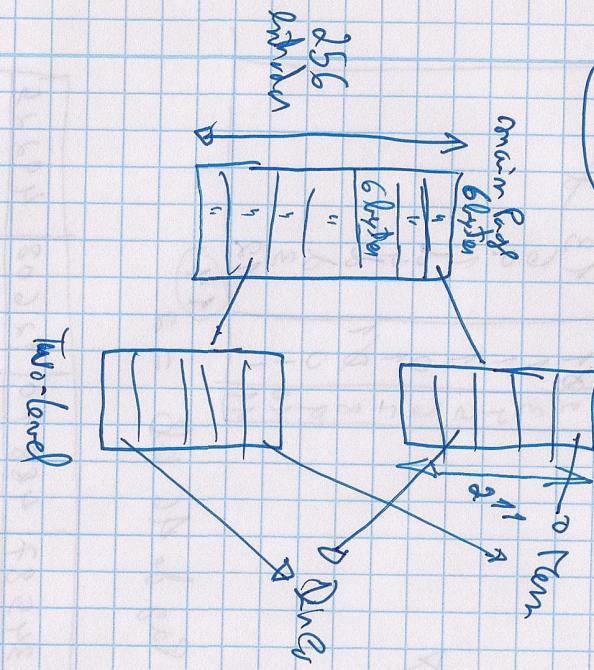
5.11.5 Paging

$$\text{Number} = \underbrace{256 \times 6}_{\text{nr 1}} + \underbrace{2^{11} \times 4}_{\text{nr 2}} \times 256 =$$

nr 2

$$1536 + 2 \text{ MB} =$$

Tables do
reservando



Note: ~~example~~
quadra 'under half' metodo o
que manda 'a segundo nível'
 $\text{Total nr} = \underbrace{256 \times 6}_{\text{nr 1}} + \underbrace{2^{11} \times 4}_{\text{nr 2}} \times 256$

$$\text{nr 1} \\ \text{nr 2}$$

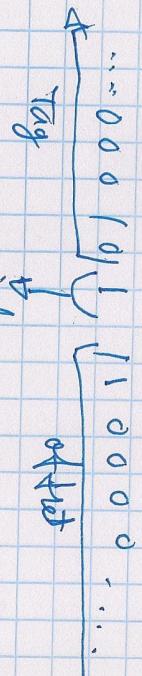
8	11	13
1 ^o nível	2 ^o nível	

5.11.3

Tag/Index	0/1	0/0	1/1	1/0	5/1	1/1	6/0
Endreier (Bpte)	4 66 9	2 22 7	1 39 16	3 45 8 7	4 88 2 0	1 260 8 4 922 5	
Page Virtual	1	0	3	8	1	11	3
Page Table	Min TLB	H.J.TLB	Min TLB	H.J.TLB	Min TLB	H.J.TLB	Min TLB
Page Frame	13	5	6	14	12	6	15
Endreier frame							

T LB	Tag	Page Frame	V	Tag	Page Frame
0	0 1	2 4	1 14	10 0 6	8 15
1	1 1	1 2	6	1	3 0 5 4 13 12

Nota: Sonderfall:



V	Page Fault during
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

→ (Page Direct) Index ist eine weitere Ebene.

Tag/Index	0/01	0/0	0/3	2/0	2/3	0/3	0/0
Endreier (Bpte)	4 66 9	2 22 7	1 39 16	3 45 8 7	4 88 2 0	1 260 8 4 922 5	
Page Virtual	1	0	3	8	11	3	12
Page Table	Min TLB	H.J.TLB	Min TLB	H.J.TLB	Min TLB	H.J.TLB	Min TLB
Page Frame	13	5	6	12	6	15	
Endreier frame							

T LB	Tag	Page Frame	V	Page Fault during
0	1	2	3	4 88 2 0
1	0 1	0	15	1 260 8 4 922 5
2	9	10	2	1 12 4 6 12 6
3	9	10	2	1 12 4 6 12 6

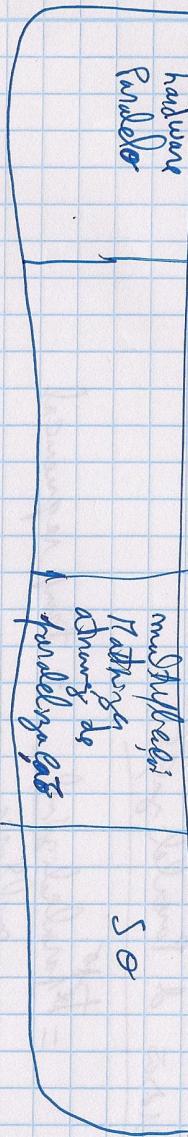
... 0 0 0 1 | 0 1 1 1 0 0 0 0 - - -

Tag index offset

19

Multitolerancia

Multitolerancia	Software	Requerimientos	Concurrente
hardware		multitolerado	soportando
Requerimientos	hardware paralelo	multiples requisitos diseño de paralelogato	SO



• thread, worker hardware parallel

• more threads est., no multiplexed demands. Concurrency

for o programs

hardware represented the program to maintain on hardware

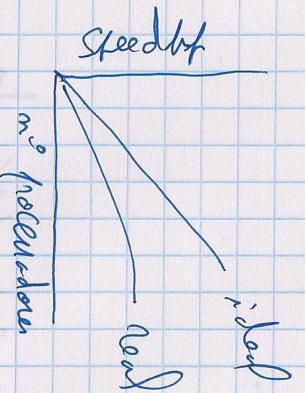
the user demand for program contain it a hardware o a

device to program depend on hardware.

Número de procesadores (NP):

$$\text{Nº de procesadores} = \frac{\text{tareas "free" paralelas}}{\text{tareas sin paralelizadas}} = NP$$

ideal



nº de procesadores

ideal

Reaktionsschritt:

$$T_{\text{depol}} = \frac{T_{\text{aktive}}}{m_{\text{effektiv}}} \approx T_{\text{aktive}} \text{ mit } m_{\text{effektiv}}$$

Versus & Parallelzonen:

$$= \frac{\text{tempo parallelzonen}}{\text{melloria}} + \text{tempo segmenten}$$

melloria

Etwas : Wenn 100 Prozenten 2 Prozenten nur Melioris $\Rightarrow = 90$?
 Wird mit $\frac{1}{2} \cdot 90$ segmentiert 90% der Effekte

$$\frac{\text{tempo aktiver}}{\text{tempo depon}} = \frac{\text{tempo aktiver}}{\frac{\text{tempo parallelzonen}}{\text{melloria}} + \text{tempo segmenten}}$$

$$\frac{\text{tempo aktiver}}{\text{tempo depon}} = 1 \quad \Rightarrow \quad = 90 \quad (\Rightarrow)$$

$$1 - \frac{\text{tempo segmenten}}{\text{melloria}} + \text{tempo segmenten}$$

$$\delta = 1 \quad \frac{1}{1 - \frac{1}{100} + \frac{1}{100}} = 90 \quad \Rightarrow \quad \frac{100}{99} - 1 = 0,1\%$$

$$1 - \frac{1}{100} + \frac{1}{100} =$$

Ambivalenz \Rightarrow :

→ Fort

→ Fort - andere kann problem weniger von min. Melioris

$$1 - \sum_{i=1}^{10} \sqrt{C_i} \quad \text{for. rep.}$$

$$2) A[10 \times 10] = B[10 \times 10] + C[10 \times 10] \quad t, \text{ for}$$

for (int $i = 0; i < 10; i++$)

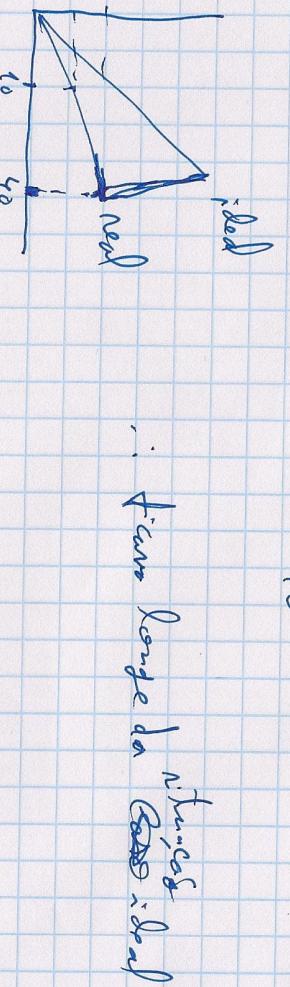
for (int $j = 0; j < 10; j++$)

$$A[C:j][i] = B[C:j][i] + C[C:i][i]$$

$$\text{Temp 1: processador} = 10t + 100t = 110t$$

$$\text{Temp 10 processador} = 10t + \frac{100t}{10} = 20t \quad \text{Media} = \frac{110}{10} = 11t$$

$$\text{Temp 40 processador} = 10t + \frac{100t}{40} = 12,5t \quad \text{Media} = 8,18$$



\therefore Fica longe da ~~memória~~ memória ideal

[outro exemplo] $N = 20$

$$\text{from } T_1 P = 10t + 400t = 410t$$

$$T_{10} P = 10t + \frac{400t}{10} = 50t \quad \text{Media} = \frac{410}{10} = 41,2$$

$$T_{40} P = 10t + \frac{400t}{40} = 20t \quad \text{Media} = 20,5$$

out, example

$$NP = 10 \text{ frekvenser}$$

$$1/nsekunder = 5\%$$

$$t_{NP} = f_0 t + f_0 \frac{5}{100} t = \cancel{f_0 t} \quad \uparrow^{100\%}$$

$$t_{NP} = 10 t + \max \left(5\%, \frac{95\%}{9} t \right)$$

$$= 10t + \max (5t, 10, 6t)$$

$$= 20,6t$$

Arquitectura de Sistemas e Computadores II

1ª Frequência

Departamento de Informática
Universidade de Évora

7 de Novembro de 2013

Indique todos os cálculos efectuados

Desempenho

1. [3 valores] Num computador, cujo relógio funciona à frequência de 2 GHz, as instruções dividem-se em três classes, cada uma com o seu CPI:

Classe	A	B	C
CPI	1	2	3

A execução do código produzido pelo compilador X para um programa leva à execução de 400 milhões de instruções, sendo 30% delas da classe A, 40% da classe B e 30% da classe C. Na execução do código gerado para o mesmo programa pelo compilador Y, são executadas só 300 milhões de instruções, distribuídas do seguinte modo pelas três classes: 20% da A, 30% da B e as restantes da C.

Qual o compilador que gera o código mais rápido para o programa? E qual o *speedup* obtido com o uso desse compilador em relação ao outro?

Implementação MIPS monociclo

Para este grupo, use como referência a implementação monociclo da Figura 1.

2. Pretende-se incluir a instrução *jr* (*jump register*) na implementação MIPS monociclo. A instrução *jr* é uma instrução tipo-R com um argumento

jr rs bits 0 | rs | 0 | 0 | 0 | 8

Esta instrução altera o fluxo de execução para a instrução cujo endereço está no registo rs.

(a) [2,5 valores] Quais das unidades funcionais existentes serão usadas para a execução desta instrução e que unidades funcionais é necessário acrescentar?

(b) [2,5 valores] Que sinais de controlo é necessário acrescentar e quais os valores que os vários sinais de controlo deverão ter para a execução desta instrução?

(Não precisa de indicar o valor de ALUOp, basta dizer qual será a função da ALU na execução desta instrução.)

Se considerar necessário fazer alguma alteração ao caminho de dados, apresenta-a na Figura 1.

Pipeline MIPS de 5 andares

Para este grupo, use como referência o *pipeline* da Figura 2. Tenha, no entanto, em atenção as caracterizações do funcionamento do *pipeline* feitas nas várias alíneas.

3. [1,5 valores] Considere, agora, a inclusão da instrução *jr* na implementação *pipelined* do MIPS. Se o valor no registo *rs* for escrito no PC no fim do ciclo de relógio em que a instrução esteve no andar EX, quantos *delay slots* deverão ser usados para o processador não ter de introduzir qualquer atraso no *pipeline*? Porquê?

4. Pretende-se executar o código MIPS seguinte de modo a que o seu efeito seja exactamente o que teria se fosse executado na implementação monociclo do processador. No fim da execução do código, os valores nos registos usados não são importantes.

1.	and	\$7, \$0, \$0
2.	beq	\$5, \$0, fim
3.	ciclo:	lw \$8, 0(\$4)
4.	add	\$7, \$7, \$8
5.	addiu	\$4, \$4, 4
6.	addi	\$5, \$5, -1
7.	bne	\$5, \$0, ciclo
8.	fim:	sw \$7, 0(\$6)

- (a) [2 valores] Identifique as dependências de dados existentes no código apresentado.

- (b) [2 valores] Simule a execução do código quando só é executada uma iteração do ciclo (a execução termina tendo a instrução 7. sido executada uma única vez). Considere um processador com *forwarding*, com decisão dos saltos condicionais no andar EX, sem *delay slots* e com previsão perfeita do resultado das instruções de salto condicional. Apresente a evolução do estado do *pipeline* durante a execução, indicando todos os atrasos introduzidos e todos os pontos onde foi necessário o *forwarding* de algum valor, identificando claramente entre que andares o *forwarding* foi feito.

Quantos ciclos de relógio são necessários para executar o código nas condições acima?
Quantos ciclos de relógio seriam necessários para executar o código se fossem executadas 100 iterações do ciclo?

- (c) [2 valores] Altere o código apresentado, reordenando as instruções e, se considerar útil, modificando o *offset* das instruções de acesso à memória, de modo a eliminar o maior número possível de atrasos durante a sua execução no *pipeline* com *forwarding*, com decisão dos saltos condicionais no andar ID e com um *branch delay slot*.

5. As latências das várias componentes do *pipeline* são as apresentadas na tabela seguinte:

PC/Registos do pipeline	Memória instruções	Banco registos	ALU	Memória dados	Somadores	Multiplexors	Controlo	Controlo da ALU
10 ps	400 ps	170 ps	330 ps	450 ps	250 ps	30 ps	40 ps	20 ps

Considere que os restantes elementos lógicos têm latência zero.

- (a) [1,5 valores] Nestas condições, qual será a frequência máxima a que processador poderá funcionar?
(b) [1,5 valores] Quais as unidades funcionais, de controlo, etc., que se encontram no caminho crítico do andar EX? Qual a latência desse caminho?

6. [1,5 valores] Diga, justificando, se será possível um *issue packet* do *pipeline* MIPS com *double issue* estático ser constituído pelas duas instruções

$$\text{add } \$1, \$2, \$3 \quad \text{e} \quad \text{lw } \$1, 0(\$4).$$

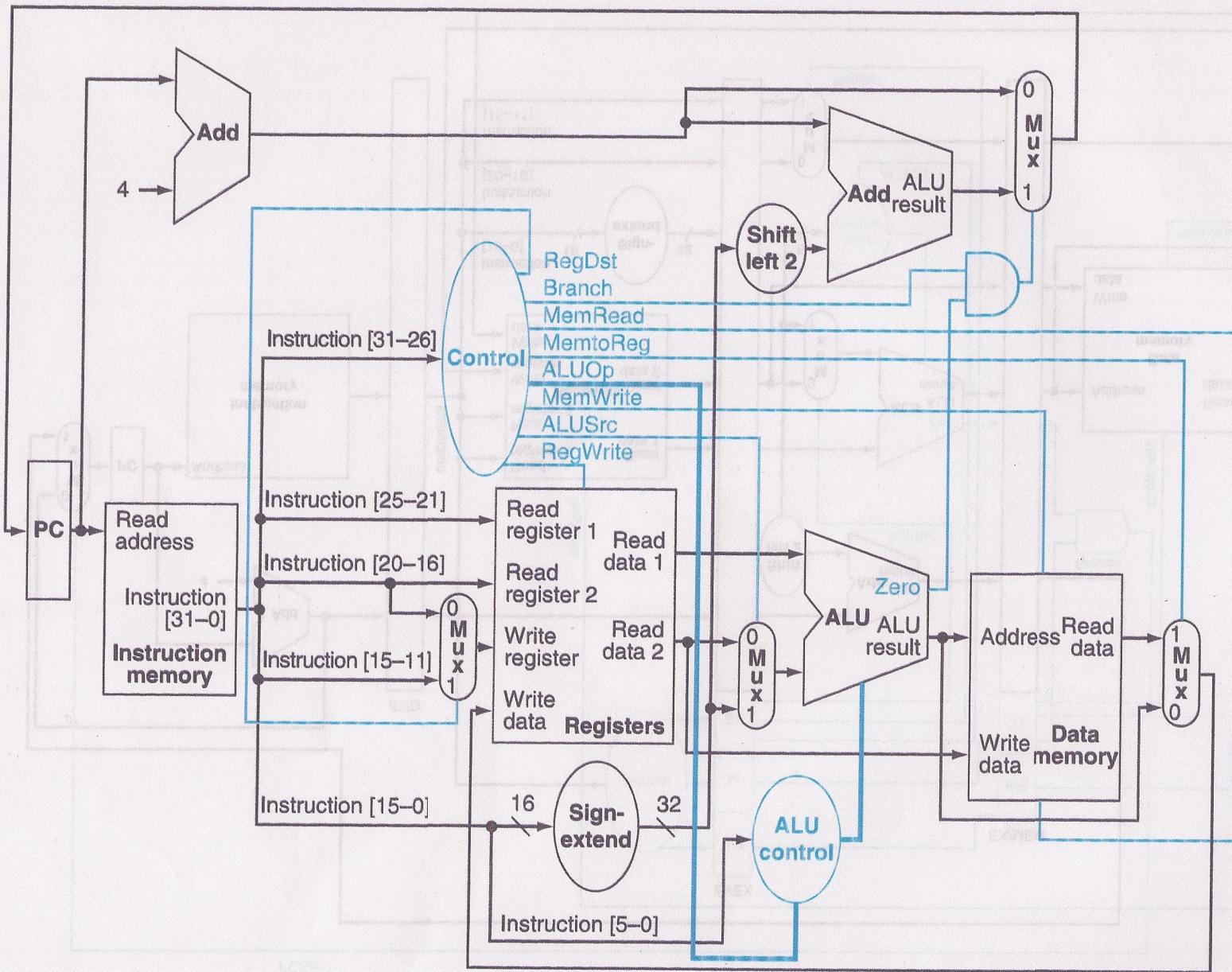
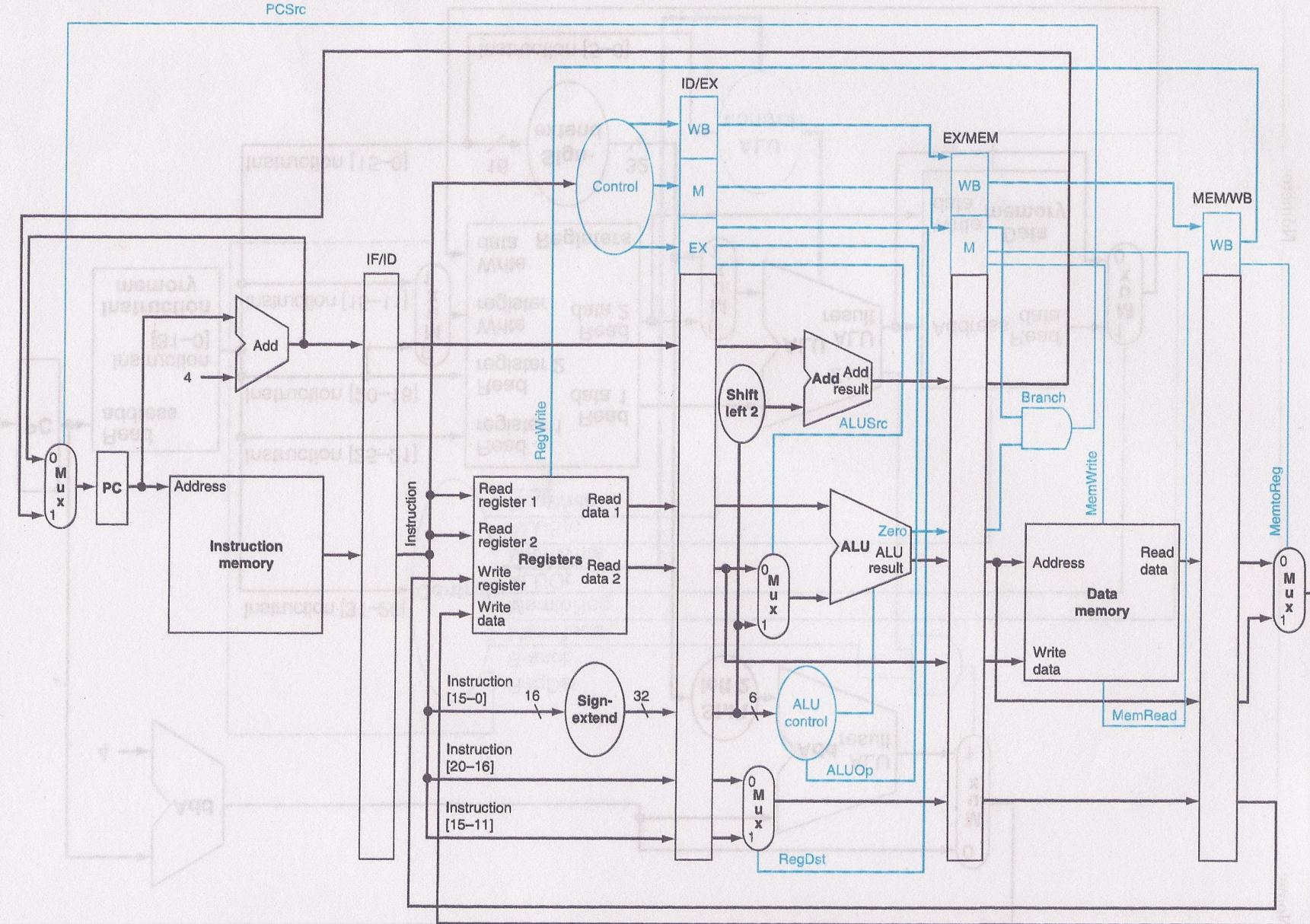


Figura 1: Diagrama de blocos da implementação MIPS monociclo

Figure 2: Diagrama de bloques do pipeline MFTS



Desempenho

$$F = 26H_3 = 2 \times 10^9$$

$$N_{inst} = 400 \text{ milhões} = 4 \times 10^8 \text{ inst.}$$

$$A - 30\% = 400 \times 0,30 = 120 \text{ milhões inst.}$$

$$B - 40\% = 400 \times 0,40 = 160 \text{ milhões inst.}$$

$$C - 30\% = 400 \times 0,30 = 120 \text{ milhões inst.}$$

$$N_{inst} = 300 \text{ milhões}$$

$$A - 20\% = 60 \text{ milhões inst.}$$

$$B - 30\% = 90 \text{ milhões inst.}$$

$$C - 300 - 60 - 90 = 150 \text{ milhões inst.}$$

Qual o mais rápido?

$$N_{ciclos_X} = N_{inst} \times CPI_A (=) N_{ciclos} = 120 \times 1 = 120$$

$$N_{ciclos_B} = N_{inst} \times CPI_B (=) N_{ciclos} = 160 \times 2 = 320$$

$$N_{ciclos_C} = N_{inst} \times CPI_C (=) N_{ciclos} = 100 \times 3 = 360$$

$$N_{ciclos_Y} = \sum [120 + 320 + 360] = 800 \text{ ciclos}$$

$$T_{exec_X} = N_{ciclos_X} / F = 800 / 2 = 400$$

$$T_{exec_Y} = N_{ciclos_Y} / F = 690 / 2 = 345$$

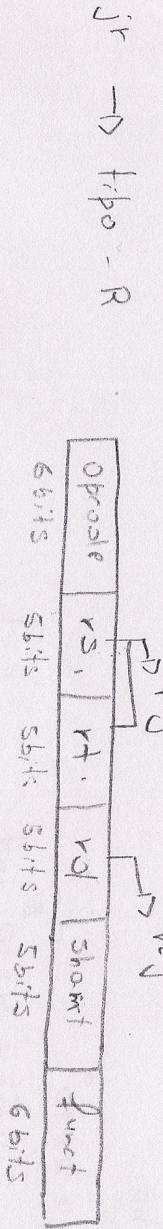
Quantas vezes mais rápido é o compilador Y em relações a X?

$$\frac{\text{Desempenho}_Y}{\text{Desempenho}_X} = \frac{T_{exec_X}}{T_{exec_Y}} = \frac{400}{345} = 1,15$$

O compilador Y é 1,15 vezes mais rápido do que o compilador X para este programa.

Implementações Mips monociclo

2.



$JR rs$

0	rs	0	0	0	8
---	----	---	---	---	---

$R = 001000_{(2)}$

- a)
- efetua um "salto" para o endereço contido no registro rs.

$$PC = ps$$

Unidades funcionais existentes que utiliza:

- Banco de registradores
- Memória de instruções
- Sign-extend
- shift-left²
- Add ALU e Mux do Add

Não é necessário acrescentar unidades funcionais.

b)

- RegDst - 0
 Branch - 0
 MemRead - 0
 MemToReg - 0
 MemWrite - 0
 ALUSrc - 1
 RegWrite - 0
- O endereço destino de salto é conhecido
 estando presente instrução JR (salto relativo)
 como um registrador.
- O opêra ALU faz nessa operação o
 verificar se o registrador rs é diferente do
 Registrador rt, mas o valor do operando
 é descartado pois não é alterado nenhum
 valor de um registo e também não é escrito
 nado na memória de dados.

$$① \bar{F}e = 2G H^2 = 2 \times 10^{-8} \text{ Hz}$$

$$m_{\text{min}} = 4 \times 10^{-8} \text{ g/g}$$

classe	A	B	c
CPT	1	2	3
P1@	30%	40%	30%
P2 Y	20%	30%	50%

classe	A	B	c
P1@	30%	40%	30%
P2 Y	$0,6 \times 10^8$	$0,9 \times 10^8$	$1,5 \times 10^8$

$$CPT \text{ global} = 8 \times 10^8$$

$$\text{CPT} = \frac{\text{N} \times \text{CPT}_e}{Fe}$$

$$\text{medio} = \frac{(1,20 \times 10^8 \times 1) + (1,6 \times 10^8 \times 2) + (1,20 \times 10^8 \times 3)}{6} = 8 \times 10^8 \text{ CPT}$$

$$\text{medio} = (0,6 \times 10^8 \times 1) + (0,9 \times 10^8 \times 2) + (1,5 \times 10^8 \times 3) = 6,9 \times 10^8 \text{ CPT}$$

$$\text{Tempo de execução} = \frac{\text{medio}}{\text{velocidade}} = \frac{8 \times 10^8}{6,9 \times 10^8} = 1,14 \text{ min} = 4 \times 10^9 \text{ s}$$

$$\text{Tempo de execução} = \frac{6,9 \times 10^8}{6,9 \times 10^8} = 2,09 \times 10^9 \text{ s} = 3,45 \times 10^9 \text{ s}$$

$$\text{tempo de execução} = \frac{6,9 \times 10^8}{2 \times 10^9} = 0,345 \text{ min} = 20 \times 10^9 \text{ s}$$

$$CPT \text{ global} = \frac{6,9 \times 10^8}{0,345} = 19,95 \times 10^8$$

$$\text{Sobrem} = \frac{3,20 \times 10^8}{1,035 \times 10^8} = 3,09 \text{ m}$$

①

Confidencial X
R2 Confidencial Y

estimado

~~$$\text{medida} = (0,30 \times 1) + (0,40 \times 2) + (0,30 \times 3)$$~~

$$\frac{1}{2} \quad 2$$

$$F.O. = 2 \cdot 6 \cdot 10^2 = 2 \times 10^9$$

<input type="checkbox"/> Confidencial X	$A = 4 \times 10^8 \times 0,3 = 1,20 \times 10^8$	<input checked="" type="checkbox"/> Imputación $\times = 1,20 \times 10^8 \times 1 + (1,60 \times 10^8 \times 2) + (1,20 \times 10^8 \times 3)$
B = $4 \times 10^8 \times 0,40 = 1,60 \times 10^8$		
C = $4 \times 10^8 \times 0,30 = 1,20 \times 10^8$		

$$= 1,20 \times 10^8 + 2,20 \times 10^8 + 3,60 \times 10^8$$

$$= 8 \times 10^8$$

Confidencial Y

$$A = 3 \times 10^8 \times 0,20 = 0,60 \times 10^8$$

$$B = 3 \times 10^8 \times 0,30 = 0,90 \times 10^8$$

$$C = 3 \times 10^8 \times 0,50 = 1,50 \times 10^8$$

$$\text{Imputación CP}_Y =$$

$$\begin{aligned} & \text{a.c.dl}_X = (0,60 \times 10^8 \times 1) + (0,90 \times 10^8 \times 2) + (1,50 \times 10^8 \times 3) \\ & = 0,60 \times 10^8 + 1,80 \times 10^8 + 4,50 \times 10^8 \\ & = 6,9 \times 10^8 \end{aligned}$$

$$\text{Temp de encuestado} = \frac{8 \times 10^8}{2 \times 10^9} = 0,4 \text{ segundos}$$

$$\text{Temp de examen} = \frac{6,9 \times 10^8}{2 \times 10^9} = 0,345$$

$$\text{Medida} = \frac{0,4}{0,345} = 1,15 \quad \text{Logo o confidencial Y } \rightarrow 1,15 \text{ segs, +}$$

10% de error para confidencial X

$$\frac{120}{120} \quad \frac{120}{120}$$

c) and \$7, \$0, \$0

bed \$5, \$0, fim

ciclo: $LW \$8,0(\$4)$

addiu \$4, \$4, 4

addi \$5, \$5, -1

add \$7, \$7, \$8

nop
 $bne \$5, \$0, \text{ciclo}$

fim: $SW \$7,0(\$6)$

fim: $SW \$7,0(\$6)$

5.

a) Memoria inst + PC + Banco Registras + ALU + Memória Dados + $(3 \times MUX)$ + controlo

$$= 400 + 10 + 170 + 330 + 450 + (3 \times 30) + 40 = 1490 \text{ ps}$$

b)

ALU \leftarrow MUX

$$330 + 30 = 360$$

6.

odd \$1, \$2, \$3

$LW \$1, 0(\$4)$

0	\$2	\$3	\$1	add
LW	\$4	\$1	imm	

Ind.	Inst.	C1	C2	C3	C4	C5	C6	C7
n	odd	IF	ID	EX	MEM	WB		
+4	Load	IF	ID	EX	MEM	WB		
+8	odd	IF	ID	EX	MEM	WB		
+12	Load	IF	ID	EX	MEM	WB		
+16	odd	IF	ID	EX	MEM	WB		
+20	Load	IF	ID	EX	MEM	WB		

Conso a instrucao odd
não produz nenhuma valor
para a instrucao LW
é possivel.